Documentație Proiect Radiali

Ratiu Miruna, Bojan Carina si Sburlea Radu

Data predarii: 16.01.2025

1 Prezentare Generala a Proiectului

Acest proiect reprezinta o aplicatie web moderna construita cu ajutorul Django, avand ca obiectiv principal oferirea unei game variate de produse care se adreseaza persoanelor care se specializeaza in industria constructiilor.
Aplicatia dispune de o interfata intuitiva, in parte de sus a site-ului se poate observa o bara de navigatie care contine urmatoarele campuri: "Home", "Despre", "Produse", "Contact" si "Intra in cont", dupa ce se executa operatiunea de "log-in" respectiv "sign-up" apar alte campuri in bara de navigatie cum ar fi "Logout" si "Cosul meu". Prin arhitectura bazata pe componente, fiecare functionalitate este modulara si usor de intretinut, permitand in acelasi timp o integrare rapida de noi caracteristici.

2 Functionalitati Principale

- **Navigare in catalogul de produse**: utilizatorii pot explora o gama de produse si componente pentru diferite utilaje de constructii.
- **Adaugarea produselor in cos**: orice produs poate fi adaugat in cosul de cumparaturi cu cantitatea dorita mai apoi apelandu-se la furnizor pentru a discuta si negocia pretul.
- **Deplasarea intre pagini**: site-ul este structurat si impartit astfel incat utilizatorului sa ii fie cat mai usor sa il utilizeze si sa gaseasca produsele de interes .
- **Pagina Acasa**: aceasta ofera o prima intalnire cu site-ul intr-un mod estetic, fiind intampinat de un carusel de poze potrivite tematicii alese.

3 Stack Tehnologic

- **Djanjo**: Django este un framework web open-source, scris in Python, care faciliteaza dezvoltarea rapida si usoara a aplicatiilor web robuste si scalabile. Este unul dintre cele mai populare framework-uri pentru Python datorita simplitatii, flexibilitatii si instrumentelor sale puternice.
- **JavaScript**: JavaScript este un limbaj de programare folosit atat pentru frontend, cat si pentru backend, datorita motorului Node.js, care permite rularea codului JavaScript pe servere. Este cunoscut pentru flexibilitate si performanta, fiind utilizat in crearea aplicatiilor web dinamice. JavaScript gestioneaza serverul si logica backend in cadrul proiectului realizat, in timp ce Django este folosit ca suport sau pentru alte parti ale aplicatiei, cum ar fi controlul functionaitatilor de tip "Admin".
- **HTML si CSS**: HTML este limbajul de marcare folosit pentru a structura si organiza continutul unei pagini web. Este baza oricarui site web, definind elementele care compun o pagina, cum ar fi texte, imagini, linkuri, tabele si multe altele. CSS este limbajul folosit pentru a stiliza elementele definite in HTML. El controleaza aspectul vizual al paginii, cum ar fi culorile, fonturile, spatierea si pozitionarea elementelor.

4 Design Patterns

4.1 Arhitectura Bazata pe Componente

Aplicatia urmeaza principiile React, fragmentand interfata in componente reutilizabile, clar definite:

- **NavBar**: Sectiune de navigare globala, prezenta pe toate paginile.
- **Pagina Principala (Acasa)**: Listeaza prosusele ordonate pe categorii si functionalitate, produsele pot fi accesate pentru mai multe detalii precum fisa tehnica. In partea de jos a paginii web se poate observa o sectiune cu detaliile de contact aale companiei Radiali.
- **Pagina Cosul meu**: Permite vizualizarea si gestionarea produselor care se doresc a fi achizitionate.

4.2 Pattern-ul MVC

Django urmeaza modelul de design MVC (Model-View-Controller), dar foloseste propria sa conventie de denumire si structura, adesea denumita MTV (Model-Template-View). Desi denumirile sunt diferite, conceptele de baza sunt strans aliniate cu modelul traditional MVC. In MVC traditional, Controller-ul gestioneaza input-ul utilizatorului si actualizeaza Modelul sau View-ul dupa caz. Totusi, in Django:

- View-ul din Django (conform denumirii sale) actioneaza ca un Controller, deoarece proceseaza cererile HTTP, interactioneaza cu modelele si trimite datele catre template-uri.
- Template-ul din Django se ocupa de randarea output-ului, ceea ce este analog cu View-ul din modelul traditional MVC.

Aceasta redenumire ajuta la sublinierea faptului ca "view"-ul din Django reprezinta ceea ce utilizatorul vede, in timp ce "controller"-ul din Django este abstractizat in URL dispatcher-ul si gestionarea cererilor din cadrul framework-ului.
In React, acest concept este adesea sustinut de Context API, permitand accesul la starea globala din orice componenta, fara a fi necesara transmiterea datelor prin props. In cazul conectarii la un serviciu extern (precum The Movie Database), putem folosi Singleton pentru gestionarea cererilor, astfel incat acelasi client de API sa fie utilizat pe tot parcursul aplicatiei.

4.3 Pattern-ul Singleton

Descriere: Singleton este un pattern in care o clasa are o singura instanta globala. Django foloseste acest pattern pentru anumite componente.

- Configurarea aplicatiei: Instantele configurarilor Django (modulele din fisierul settings.py) sunt Singleton-uri.
- Conexiunea la baza de date: Django mentine o conexiune unica (singleton) pentru fiecare baza de date definita.

4.4 Pattern-ul Router

Navigarea in aplicatie este gestionata prin **React Router**, un exemplu clasic de implementare a pattern-ului „Router”. Codul esential arata astfel:

```
<Routes>
    path("cart/", views.cart_detail, name="cart_detail"),
    path("add/<int:product_id>/", views.add_to_cart, name="
       add_to_cart"),
    path("remove/<int:cart_item_id>/", views.remove_from_cart,
       name="remove_from_cart"),
    path("decrease/<int:cart_item_id>/", views.decrease_quantity,
        name="decrease_quantity"),
    path("increase/<int:cart_item_id>/", views.increase_quantity,
        name="increase_quantity"),
    path('process-order/', views.process_order, name='
       process_order'),
</Routes>
```

Acest sistem asigura rute clare si management eficient al istoricului de navigare, mentinand totodata URL-uri semnificative.

5 Arhitectura Aplicatiei

5.1 Diagrama de Secventa

Diagrama de secventa descrie procesul de autentificare al unui utilizator intr-un sistem, incluzand gestionarea cazurilor in care parola sau username-ul sunt incorecte sau lipsesc. Procesul incepe cu utilizatorul care introduce username-ul si parola in interfata utilizatorului. Datele de autentificare sunt trimise mai departe catre serverul de autentificare, care verifica daca username-ul exista in baza de date. Daca username-ul nu exista, utilizatorului i se afiseaza un mesaj prin care este indemnat sa se inregistreze (”Sign Up”).

In cazul in care username-ul este gasit, serverul verifica daca parola introdusa este corecta. Daca parola este incorecta, utilizatorului i se afiseaza un mesaj de eroare (”Parola incorecta”) si i se ofera optiunea de a reintroduce parola sau de a initia procesul de resetare a parolei. Daca utilizatorul opteaza pentru resetare, serverul genereaza o parola temporara si trimite un mesaj de confirmare, de obicei prin email. Utilizatorul primeste confirmarea ca parola a fost resetata cu succes.

Daca parola este corecta, serverul valideaza autentificarea si utilizatorul este logat in sistem, primind mesajul ”Welcome!”. Diagrama subliniaza interactiunile intre utilizator, interfata utilizator, serverul de autentificare si baza de date pentru a ilustra complet acest proces (Figura 1).

5.2 Diagrama de Activitate

Aceasta diagrama de activitate ilustreaza procesul de autentificare al unui utilizator intr-un sistem si include pasii necesari pentru a trata situatiile in care datele de autentificare sunt incorecte sau lipsesc.

Procesul incepe cu utilizatorul care introduce username-ul si parola in interfata de autentificare. Datele introduse sunt trimise catre sistem pentru verificare. Sistemul verifica

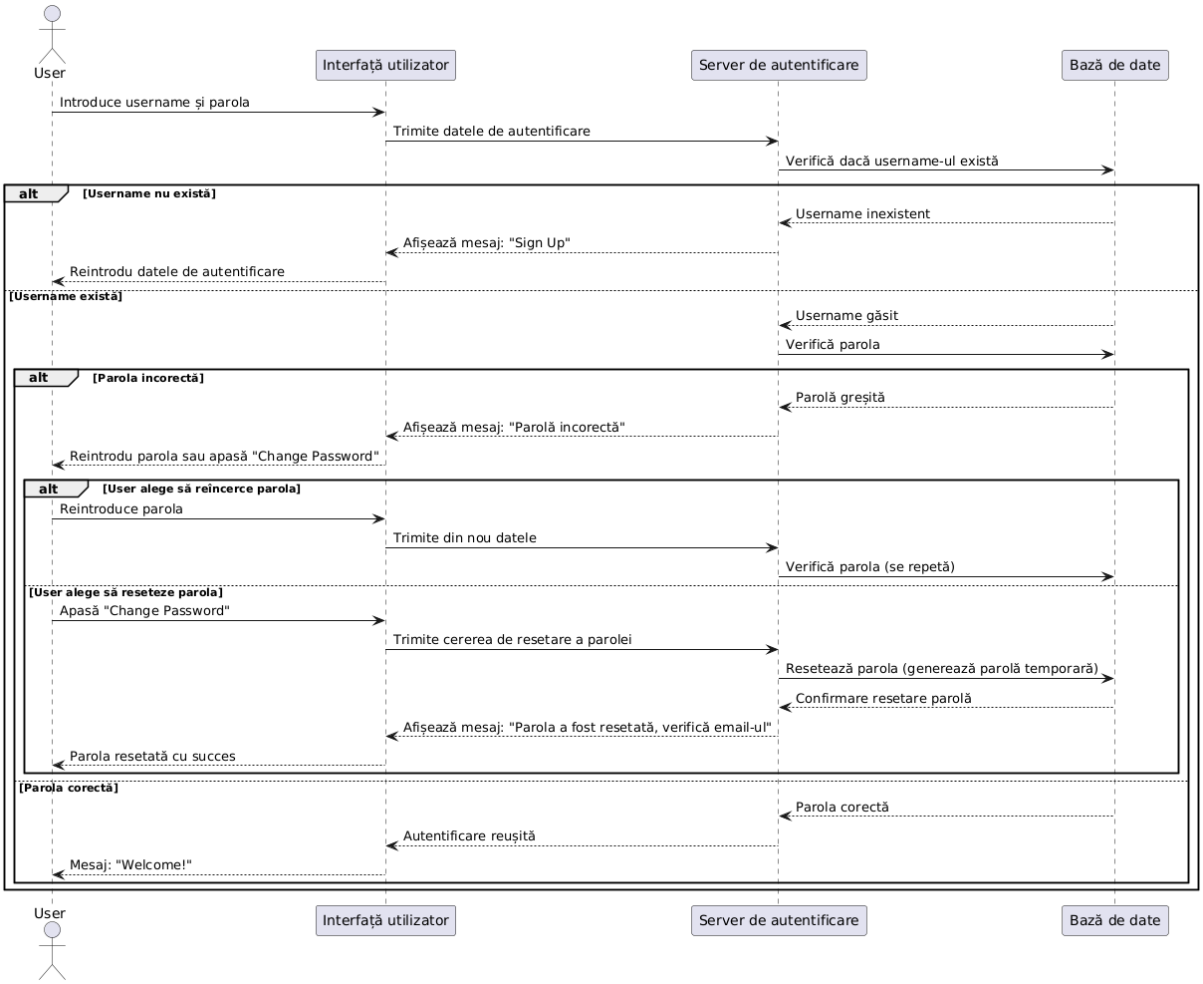

Figura 1: Diagrama de secventa Ratiu Miruna

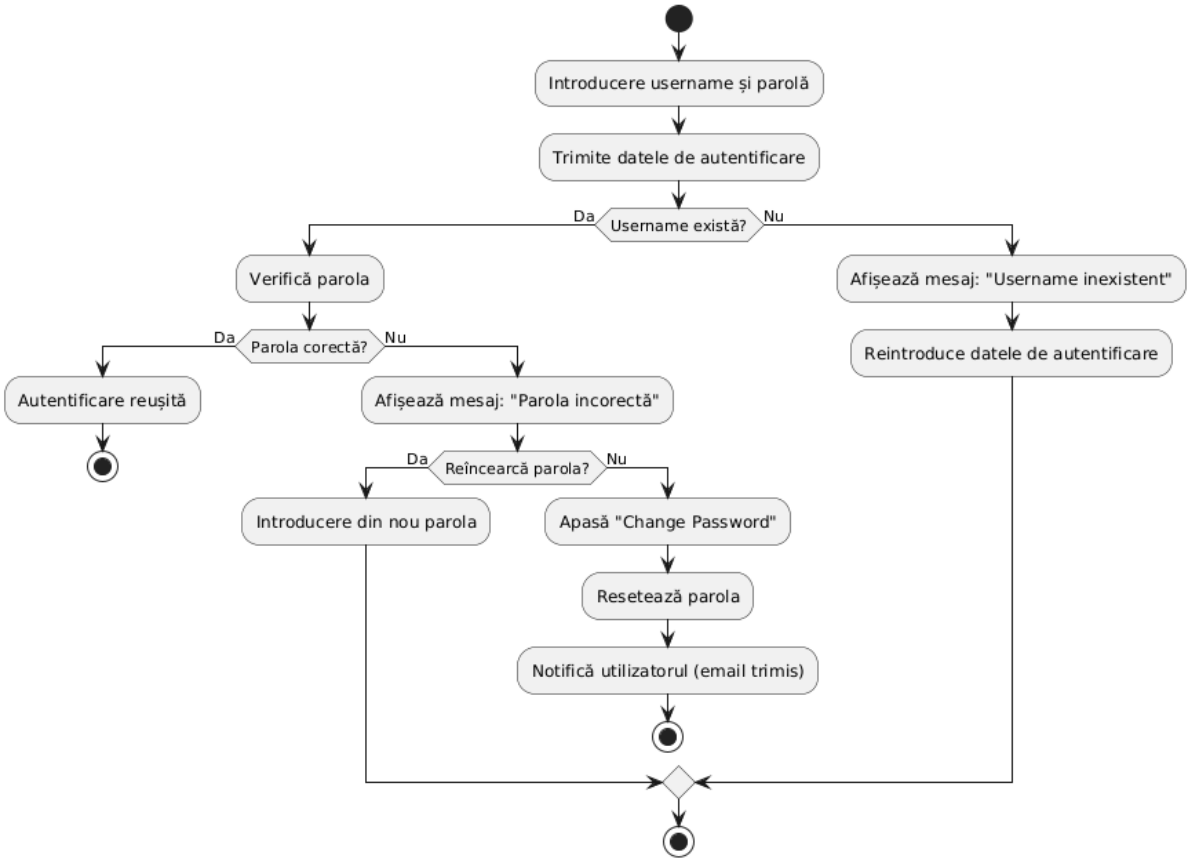

Figura 2: Diagrama de activitate Sburlea Radu

mai intai daca username-ul exista. Daca username-ul nu exista, utilizatorului i se afiseaza un mesaj ("Username inexistent") si i se cere sa reintroduca datele de autentificare.

Daca username-ul este gasit, sistemul verifica daca parola este corecta. In cazul in care parola este corecta, autentificarea este reusita, iar procesul se incheie.

Daca parola este incorecta, utilizatorului i se afiseaza un mesaj ("Parola incorecta") si i se ofera doua optiuni: sa incerce din nou sa introduca parola sau sa opteze pentru resetarea parolei.

Daca utilizatorul alege sa reintroduca parola, procesul revine la verificarea parolei. Daca utilizatorul alege sa reseteze parola, acesta apasa pe optiunea "Change Password", iar sistemul genereaza o parola temporara. Utilizatorul este notificat despre resetarea parolei, de obicei prin email.

Procesul se incheie dupa resetarea parolei sau dupa ce utilizatorul introduce o parola corecta, finalizand autentificarea. Diagrama subliniaza toate ramificatiile posibile in procesul de autentificare si trateaza cazurile de eroare (Figura 2).

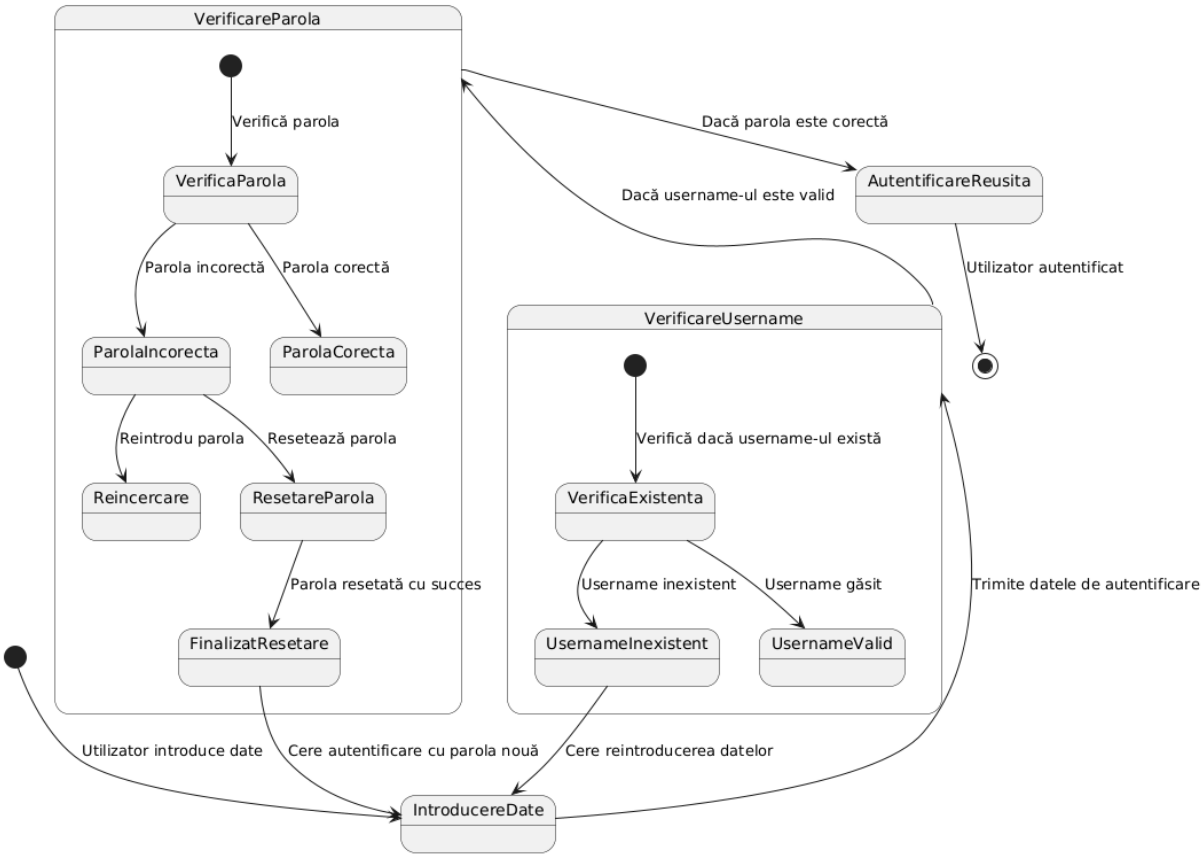

Figura 3: Diagrama de stare Bojan Carina

5.3 Diagrama de Stare

Aceasta diagrama de stare descrie procesul de autentificare al unui utilizator, incluzand verificarile pentru username si parola, precum si actiunile asociate diverselor situatii. Procesul incepe cu starea de IntroducereDate, unde utilizatorul introduce username-ul si parola. Datele sunt trimise mai departe catre sistem pentru verificare.

In modulul VerificareUsername, sistemul verifica daca username-ul introdus exista. Daca username-ul nu exista, se trece la starea UsernameInexistent, iar utilizatorului i se

cere sa reintroduca datele. Daca username-ul este valid, procesul continua cu verificarea parolei.

In modulul VerificareParola, parola introdusa este verificata. Daca parola este corecta, sistemul trece la starea AutentificareReusita si autentifica utilizatorul cu succes. In cazul in care parola este incorecta, sistemul trece la starea ParolaIncorecta. In aceasta situatie, utilizatorului i se ofera doua optiuni: sa reintroduca parola, revenind astfel la modulul de verificare a parolei, sau sa aleaga optiunea de resetare a parolei. Daca utilizatorul alege resetarea parolei, sistemul genereaza o noua parola si trece la starea FinalizatResetare.

Dupa finalizarea resetarii parolei, utilizatorul este redirectionat inapoi la starea de introducere a datelor pentru a relua procesul de autentificare. Diagrama evidentiaza clar tranzitiile intre diferitele stari si actiunile declansate in functie de rezultatul verificarilor efectuate, acoperind toate cazurile posibile in procesul de autentificare (Figura 3).

6 Detalii de Implementare

Managementul produselor include un model Product care contine campuri pentru nume, pret, stoc, imagine si fisier tehnic. Sistemul permite cautarea produselor, filtrarea lor pe baza categoriilor si implementarea unui sistem de categorii direct in modelul Product.

Frontend-ul este bazat pe un grid de produse implementat in JavaScript (productsGrid.js) si dispune de o interfata responsiva pentru afisarea produselor. Sistemul de incarcare lazy pentru imagini asigura optimizarea performantelor, iar functionalitatile interactive faciliteaza gestionarea produselor.

Din punct de vedere tehnic, proiectul utilizeaza Django pentru backend, avand migrari pentru gestionarea bazei de date. Imaginile sunt gestionate prin campuri de tip URLField, iar sistemul de rutare permite navigarea intre diferite categorii de produse. Este un proiect bine structurat, care separa logica de business in module distincte si integreaza tehnologii moderne pentru implementare.

7 Dezvoltari Viitoare

- **O noua modalitate de plata**: Cu toate ca site-ul este gandit pentru a negocia pretul produsului cu dealerul am dori pe viitor ca platile sa se poata realiza prin intermediul site-ului intr-un mod sigur.

- **Aplicatie**: In viitor am dori sa dezvoltam si o aplicatie de tip mobile pentru clientii Radiali.

8 Concluzie

Prin acest proiect, demonstram ca o structura coerenta si un design bine gandit care aduce beneficii atat in dezvoltare, cat si in intretinere. Arhitectura bazata pe componente, impreuna cu design patterns precum Singleton, MVC si ..., asigura coerenta codului si lizibilitatea acestuia. Am invatat cum sa utilizam framework-ul Django pe care l-am imbinat cu elemente de JavaScript pentru back-end si Html, respectv CSS pentru front-end si stilizarea site-ului. Impartirea pe task-uri a echipei: Sburlea Radu - implementare back-end mai putin log-in si register, Ratiu Miruna - implementare back-end Log-in si front-end, Bojan Carina - implementare back-end Register si front-end

In plus prin acest proiect am dezvoltat abilitatea de a lucra in echipa, de a utiliza GitHub-ul pentru a facilita lucrul in tandem pe mai multe fronturi ale proiectului, Front-end, Back-end respectiv baza de date.

8.1 Bibliografie

- **MySQL Database**: https://dev.mysql.com/doc/
- **React Documentation**: https://docs.djangoproject.com/en/5.1/
- **JavaScript Documentation**: https://developer.mozilla.org/en-US/docs/Web/JavaScript
- **Django Tutorial**: https://www.youtube.com/watch?v=rHux0gMZ3Egab$_channel = ProgrammingwithMosh$